"A Good Algorithm Does Not Steal – It Imitates": The Originality Report as a Means of Measuring When a Music Generation Algorithm Copies Too Much

Zongyu Yin[1][0000−0001−8709−8829], Federico Reuben[2], Susan Stepney[1], and Tom Collins[2][0000−0001−7880−5093]

[1] Department of Computer Science, University of York, York, UK
{zy728,susan.stepney}@york.ac.uk
[2] Music, Science and Technology Research Cluster, Department of Music, University of York, York, UK
{federico.reuben,tom.collins}@york.ac.uk
https://mstrcyork.org

Abstract. Research on automatic music generation lacks consideration of the originality of musical outputs, creating risks of plagiarism and/or copyright infringement. We present the originality report – a set of analyses for measuring the extent to which an algorithm copies from the input music on which it is trained. First, a baseline is constructed, determining the extent to which human composers borrow from themselves and each other in some existing music corpus. Second, we apply a similar analysis to musical outputs of runs of MAIA Markov and Music Transformer generation algorithms, and compare the results to the baseline. Third, we investigate how originality varies as a function of Transformer's training epoch. Results from the second analysis indicate that the originality of Transfomer's output is below the 95%-confidence interval of the baseline. Musicological interpretation of the analyses shows that the Transformer model obtained via the conventional stopping criteria produces single-note repetition patterns, resulting in outputs of low quality and originality, while in later training epochs, the model tends to overfit, producing copies of excerpts of input pieces. We recommend the originality report as a new means of evaluating algorithm training processes and outputs in future, and question the reported success of language-based deep learning models for music generation. Supporting materials (code, dataset) will be made available via https://osf.io/96emr/.

Keywords: Music generation · Machine learning · Originality evaluation

1 Introduction

A quotation from Igor Stravinsky reads: "A good composer does not imitate, he steals" [39]. The quotation, while made in relation to a serial work, reflects Stravinsky's general interest in incorporating melodies, harmonic language, and

forms from previous periods into new works such as his *Pulcinella Suite* (1922). Stravinsky uses the term "imitate" with a negative connotation: he would rather steal, say, a melody wholesale and rework it in a contemporary piece, than he would make mere allusions to (imitate) the work of past or contemporary composers. With respect to the current paper's context – the rise of AI music generation algorithms – we instead use the term "imitate" with a positive connotation and the term "steal" with a negative connotation. As we show, some deep learning algorithms for music generation [16] are copying chunks of original input material in their output, and we would count it as a success if an algorithm – from the deep learning literature or otherwise – could generate output that *sounded like* (imitated) – but did not *copy from* – pieces in a specific style.

Research on artificial intelligence (AI) has achieved various feats of simulating human perception (e.g., [14]) and production (e.g., [28]). A number of music generation models have been developed in recent decades, many predating or outside of deep learning [34,6] and some espousing a belief in the superiority of deep learning [29,11]. We have observed, with increasing alarm, that deep learning papers on music generation tend to rely solely or primarily on loss and accuracy as a means of evaluation [16,29]. If there are listening studies, they employ listeners with inadequate expertise, and there is little or no musicological analysis of outputs, and no analysis of whether generated material plagiarises (steals from) the training data. As an increasing number of musicians are now incorporating AI into their creative workflows, checking an AI's output for plagiarism is now a paramount challenge in this area. To this end, this paper considers the topic of **automatic stylistic composition** – a branch of automatic music generation where there is a stated stylistic aim with regards to the algorithm output, and a corpus of existing pieces in the target style.

In this context, we aim to establish a framework for checking the originality of auto-generated music with a specified style. We introduce and exemplify the originality report as a means of measuring when a music generation algorithm copies too much. We discuss how to calculate a distribution for the extent to which human composers borrow from themselves or each other in some corpus of pieces in a specific style; then we discuss how to use this as a baseline while moving a sliding window across a generated passage and measuring originality as a function of time in the generated material, complementing this with a musicological analysis of outputs from prominent deep [16] and non-deep [6] learning models. Finally, for the deep learning model, we interrogate how originality varies with the training epoch.

2 Related Work

2.1 Music plagiarism

Music plagiarism is said to have occurred when there is demonstrable and perceivable similarity between two songs or pieces of music (hereafter, pieces), and when there is circumstantial evidence to indicate that the composer(s) of the latest piece would have been familiar with the existing piece. Stav [32] describes

how the musical dimensions of melody, harmony, and rhythm contribute to music plagiarism, and gives an example-based explanation of how these dimensions have been used in handling music copyright disputes. Based on the features of melodies involved in selected plagiarism cases, Müllensiefen and Pendzich [22] derive an algorithm for predicting the associated court decision, and it identifies the correct outcome with 90% success rate. Recent failed or overturned cases also indicate that while music similarity and circumstantial evidence are necessary for delivering a verdict in favour of plagiarism having occurred, they are not sufficient, in that the distinctiveness of the music with respect to some larger corpus plays an important role too [8,4,24]: melodies that share contours and begin and end on the same scale steps may well point to potential cases of plagiarism, but it is likely that other melodies will have these same characteristics too [24]; drum beats, where the initial space of possibilities is smaller compared to pitched material, have been less successful as bases for music plagiarism convictions [26].

Recently, discussions on ethical issues surrounding AI have attracted widespread attention. Collins [7] et al. use a note-counting approach to show that twenty bars of computer-generated musical output from an algorithm by Cope [10] have 63% coincidence in pitch-rhythm combinations with a piece by Frédéric Chopin. In [33], a music generation algorithm's output and its tendency to copy original input pieces motivates the posing of open questions with respect to AI and music copyright law. As such generative models learn from existing music data, the copyright status of the output is unclear. Additionally, the evaluation of these models' outputs tends to be narrow – i.e. it does not involve any kind of **originality analysis** with respect to the human-composed pieces used for training – which creates copyright or plagiarism infringement risks for musicians who are using these algorithms as part of their creative workflows.

2.2 Cognitive-computational approaches to music similarity

Largely outside of the role played by similarity in determining cases of music plagiarism, the systematic study of music similarity has a relatively long lineage [31] and continues to be of interest to scholars [37]. One challenging aspect of studying the phenomenon is that two excerpts of music can be similar to one another in myriad ways (genre, instrumentation, timbre, tempo, dynamics, texture, form, lyrics, and mentioned above, melody, harmony, rhythm). This challenge interacts with variability in use cases too. Take a single paradigm such as query-based search in the form of music identification, which relies on some implementation of music similarity. Even for this one paradigm, there are various use cases: Shazam addresses the need for exact matching [38], a variant of SoundHound addresses query-by-humming (the user sings or hums at the interface and expects "successful" results),[1] and Folk Tune Finder allows lyrics or notes to be input and, as with SoundHound's query-by-humming variant, the user's expectation of Folk Tune Finder is that the sought-after song will be found, or at least something relevant or interesting will be returned.[2] Of these

[1] https://www.midomi.com/
[2] https://www.folktunefinder.com/

use cases, only the one addressed by Shazam is clear cut – the other two are made more challenging by variation in cognitive and music-production capabilities of users, and there not necessarily being one "right answer".

Here, we are concerned with a more reductive view of music similarity – the type of note-counting approaches mentioned above. This is the characterisation of music similarity that a teacher might employ if a student's composition appears to draw too heavily on or copy directly from a known piece. For instance, "Why do 90% of the pitch-rhythm combinations in bars 1–20 of your piece occur also in this string quartet movement by Haydn?!" The representations and calculations required to reason this way, especially in algorithmic fashion, began in [19] and have been implemented in various forms since [35,1,4]. In the note-counting vein, in Section 3.1 we define a similarity measure based on the P3 algorithm [35]. We finish this section of the review with some remarks about choices of music representation and comparison methods.

Generally, researchers take **sequential** (e.g., [8,9]) or **geometric** (e.g., [21,4]) approaches to the representation and comparison of music. There are pros and cons to each approach. With the sequential approach, if one chooses to focus on MIDI note number alone and two melodies have the same MIDI notes (up to transposition) but different rhythms, a sequential representation (specifically, difference calculations between consecutive notes) will recognise these melodies as similar, whereas a geometric representation may not. However, with a sequential representation, it is less obvious how to handle polyphony (multiple notes beginning and ending at possibly different times), whereas a geometric representation can encode a polyphonic piece as easily as it encodes a monophonic piece. For instance, in the sequential representation shown in Fig. 1(d) (which is Music Transformer's [16] chosen input representation, see next section), the tokens encoding the occurrence of the F♯4 and second F♯5 are ten indices apart, even though the notes sound together. So any parameter that allows these events to be recognised as related has to be large enough to span this gap in indices. Moreover, an embellished (or, on the other hand, reduced) variation of some melody may not be recognised by the sequential representation as similar because the relationships between adjacent notes will be altered by the added or removed notes, even though the "melodic scaffold" remains intact. A geometric representation may be more robust to this kind of variation.

2.3 Music generation models

Recently, a large number of deep learning models have been proposed for symbolic music generation [29,16,12]. Several of them regard music as a sequence of tokens, where generation involves predicting the next token based on previous tokens [29,16]. Oore et al. [25] introduce a way to serialise polyphonic music and apply recurrent neural networks (RNNs) to generate output with expressive timing and velocity (loudness) levels. Huang et al. [16] use this same serialisation to adapt a transformer model [36] to generating music. Benefiting from the self-attention mechanism, it achieved lower validation loss compared to

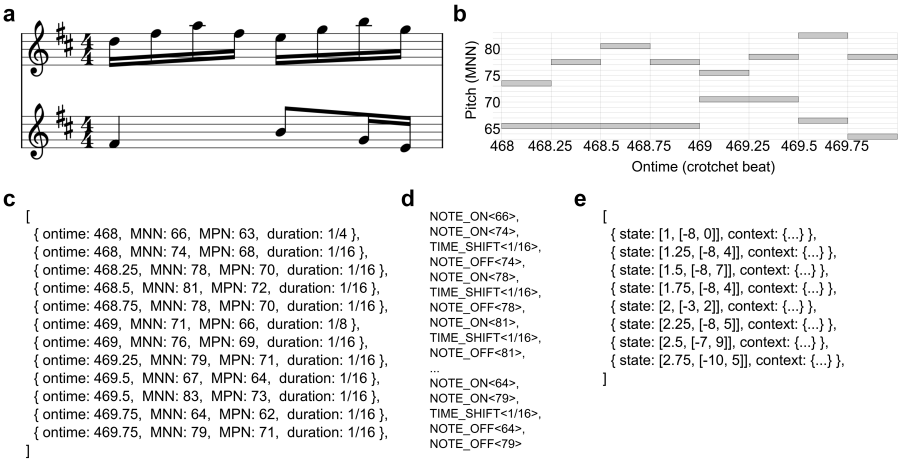

Fig. 1. Examples of symbolic music representations, starting from the same excerpt. (a) half a bar of music; (b) the so-called piano-roll representation indicating some of the music's numeric properties; (c) a 4-dimensional representation of the music as a set of points; (d) one sequential representation that handles polyphony; (e) another sequential representation that handles polyphony.

the RNN of [25] and also longer-term stylistic consistency than previous RNNs-based approaches. In other work, based on the assumption that each musical output can be sampled from a normal distribution, [29] use variational autoencoders (VAEs) combined with long short-term memory networks (LSTMs). The application of generative adversarial networks (GANs) and convolutional neural networks (CNNs) to music generation has been explored also [12], using the piano-roll representation as in Fig. 1(b) and treating music as images that can be generated in a hierarchical manner.

An issue with all the above deep learning approaches to music generation is that there has been inadequate consideration of music plagiarism in the algorithms' outputs. One user of the Music Transformer algorithm, Ruiz, writes:

> The thing is that I ran the code on my machine and it overfits. It needs a way to check that it isn't stealing from the dataset say no more than 6 or 8 continuous notes. If it can't do that it's useless. I mean your piano dataset is huge but after running the program for 20 times I found it composes note by note music of well known classical melodies. That's not OK. That should be avoided [30].

Simon, a member of the Google Magenta team, replies:

> In the checkpoints we've released, we tried hard to reduce the ability of the model to perform pieces from the training set. And in the samples we released, we tried hard to remove any samples that are too similar

> to an existing piece of music. But it's difficult to get to 100% on these for a number of reasons, including the lack of a clear definition for "too similar" [30].

Members of the general public can make use of Music Transformer for laudable reasons – Google Magenta have open-sourced the code – but their attempt to guard against music plagiarism appears problematic, and whatever constitutes "trying hard" in the above quotation has not been open-sourced, leaving general musicians who use Magenta algorithms in their creative workflows at risk of copyright infringement.

A non-deep learning approach to music generation that uses Markov models, pattern discovery, and pattern inheritance to ensure that generated material evidences long-term, hierarchical repetitive structure, also constitutes the first use of an **originality or creativity analysis** to assess the extent to which the model plagiarises human-composed works by J.S. Bach and Chopin on which it is based [6]. This algorithm, called MAIA Markov, uses the representation given in Fig. 1(e), where each state consists of a beat of the bar and the MIDI note numbers relative to tonal centre occurring on that beat.

The remainder of this paper studies two of the most promising models for music generation, Music Transformer [16] and MAIA Markov [6], and focuses on the concept of originality, and methodologies for measuring it, which are then implemented and discussed.

3 Method

This section introduces the method we use to analyse the originality of one set of symbolically encoded music excerpts relative to another. We begin by defining the two sets of excerpts: the queries (the excerpts we are testing for originality), $\mathcal{Q}$, and the targets (the excerpts we are testing against), $\mathcal{R}$. Depending on the use case, $\mathcal{Q}$ may contain one or more excerpts from one or more pieces of music, and $\mathcal{R}$ usually contains overlapping excerpts from multiple pieces of music. The use cases for wanting to produce an **originality report**, which we explore further and exemplify, are as follows:

1. The user wants to determine the "baseline" level of originality within a corpus $\mathcal{C}$. In this instance, the queries $\mathcal{Q}$ are a (pseudo-)random sample from $\mathcal{C}$, drawn from pieces $\mathcal{Q}^*$, and the targets $\mathcal{R}$ are the set complement, $\mathcal{R} = \mathcal{C} \backslash \mathcal{Q}^*$.[3] The outcome is a sample of N originality scores, from which estimates of the underlying distribution can be made, such as mean originality and confidence intervals about this mean.

[3] We distinguish between $\mathcal{Q}$ and $\mathcal{Q}^*$ because if some excerpt $q \in \mathcal{Q}$ repeats or substantially recurs elsewhere in the piece q^* from which it is drawn, and we leave this repetition in the set of targets $\mathcal{R}$, then q would be considered trivially unoriginal. Therefore, it is sensible to hold out entire pieces from which queries are selected.

2. The user wants to determine the level of originality of an algorithm's output relative to a corpus whose contents the algorithm is designed to imitate. In this instance, the queries $\mathcal{Q}$ would be overlapping excerpts of the algorithm's output, and we plot the originality of elements of $\mathcal{Q}$ as a function of time in the output, relative to elements of the target corpus $\mathcal{R}$. As well as plotting, we could also compare the mean or minimum originality found for the algorithm output to the distribution mentioned in the previous point, in which case the distribution acts as a "baseline" and can be used to address the question, "Is this algorithm's output sufficiently original?".
3. The user wants to incorporate originality reports into the modelling process itself, in order to analyse or steer/halt that process. The details are similar to the previous point, but the deployment of the method is during training or generation rather than after the fact.

3.1 Originality, Similarity, and Set of Points

To implement the originality reports that are associated with each of the use cases, it is necessary to employ at least one similarity measure – that is, some function $c : \mathcal{Q} \times \mathcal{R} \to [0, 1]$, which takes two symbolically encoded music excerpts q and r and returns a value in the range $[0, 1]$, indicating q and r are relatively similar (value near one) or dissimilar (value near zero). The measure ought to be commutative, $c(q, r) = c(r, q)$, and have an symmetry-like property that $c(q, q) = 1$. The choice of similarity metric influences subsequent decisions with respect to addressing questions such as "Is this algorithm's output sufficiently original?", but the contributions of this paper are the delineation of the use cases and the originality reports themselves, rather than the definition or use of any one similarity metric in particular.

Each originality report centres on calculating an **originality score**, OS, for some query q in relation to the set of targets $\mathcal{R}$. In particular, we find the element $r \in \mathcal{R}$ that maximises the similarity score $c(q, r)$, and subtract it from 1:

$$OS(q, \mathcal{R}) = 1 - \max\{c(q, r) \mid r \in \mathcal{R}\} \tag{1}$$

So the originality score $OS : \mathcal{Q} \times \mathbb{P}\,\mathcal{R} \to [0, 1]$, where $\mathbb{P}\,\mathcal{R}$ is the power set of $\mathcal{R}$, is also a measure in the range $[0, 1]$ indicating q is original relative to $\mathcal{R}$ (value near one) or unoriginal (value near zero). If q is a copy of something that occurs in $\mathcal{R}$, then the originality score will give $OS(q, \mathcal{R}) = 0$.

In this paper, we use a similarity measure called the cardinality score, cs [35,5,17]. To calculate it, we represent each music excerpt as a set of points containing the start time (in crotchet beats) and numeric pitch representation (morphetic pitch number [20]) of each note.[4] So an element of the query set $\mathcal{Q} : \mathbb{P}(N \times N)$ is represented as $q = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, and an element of the target set $\mathcal{R} : \mathbb{P}(N \times N)$ is represented as $r = \{(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_{n'}, y'_{n'})\}$.

[4] We use morphetic pitch in preference to MIDI note number here because the former is robust to major/minor alterations.

An example of this representation is provided in Fig. 2(c) and (d). The bottom-left point in (c) has the value $(x_1 = 468, y_1 = 53)$, representing a start time at the beginning of the excerpt ($x_1 = 468$) and the morphetic pitch for C3 ($y_1 = 53$). The viola and cello have coincident notes at this moment, which project to a single point in our representation.

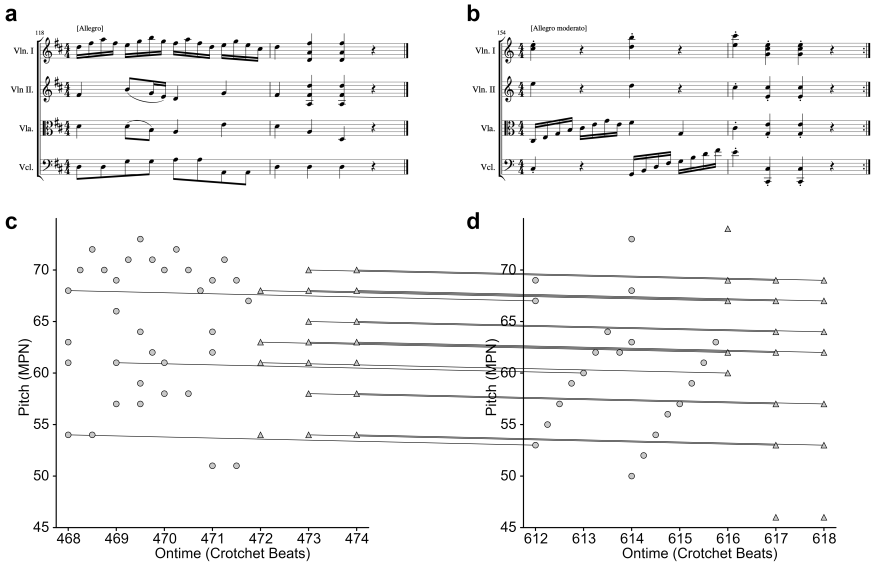

Fig. 2. Visualisation of the cardinality score between two excerpts. (a) a 2-bar excerpt from Mozart; (b) a 2-bar excerpt from Haydn; (c) mapping notes in the excerpt (a) to a set of points; (d) mapping notes in the excerpt (b) to a set of points. For clarity, notes in the first/second bars are shown as circles/triangles.

Letting $\boldsymbol{t}$ be the translation vector that gives rise to the maximum cardinality of the intersection $(q + \boldsymbol{t}) \cap r$, we define the **cardinality score** as

$$cs(q, r) = |(q + \boldsymbol{t}) \cap r| / \max\{|q|, |r|\} \tag{2}$$

where $|q|$ is the size of the set of points q. We demonstrate calculations of the cardinality score with reference to the examples in Fig. 2. In the top half of this figure, there are two excerpts of string quartets: (a) is by Mozart and (b) is by Haydn. Considering the set of points corresponding to bars 119 of the Mozart and 155 of the Haydn (second bars in both Figs. 2(a) and (b), with corresponding points shown as triangles), the vector $\boldsymbol{t} = (-144, -2)$ translates 15 points from the Mozart excerpt to points in the Haydn, and the more numerous of the two sets is the Haydn excerpt, with 19 points, so the cardinality score is $cs(q, r) = 15/19 \approx 0.7895$. As a second example, considering larger point sets

corresponding to bars 118-119 of the Mozart and 154-155 of the Haydn, the vector $\boldsymbol{t} = (-144, -2)$ translates 18 points from the Mozart excerpt to points in the Haydn, and the more numerous of the two sets is the Mozart excerpt, with 52 points, so the cardinality score is $cs(Q, R) = 18/52 \approx 0.3462$.

4 Originality Reports

The dataset we use in this paper contains 71 Classical string quartets in MIDI format from KernScores.[5] The dataset was formed according to the following filters and constraints:

- string quartet composed by Haydn, Mozart, or Beethoven;
- first movement;
- fast tempo, e.g., one of Moderato, Allegretto, Allegro, Vivace, or Presto.

4.1 Determining the Baseline Level of Originality Within a Corpus

To form the query and target sets, we divide the 71 excerpts into two sets: 50 queries $\mathcal{Q}$ were drawn from 7 pieces $\mathcal{Q}^*$, and the targets $\mathcal{R}$ consisted of the remaining 64 pieces. The selection of the 7 pieces was pseudo-random to reflect the representation of composers and time signatures in the overall dataset. We used a fixed window size of 16 beats for each query, and ran the code outlined in Algorithm 1.

Algorithm 1 Estimating the self-originality of a corpus

Require: $\mathcal{Q}$, $\mathcal{R}$, query and target corpus, respectively.
Initialize O to an empty output list.
Initialize N to the number of originality scores.
for $(i \coloneqq 0, i < N, i++)$ **do**
    $q \coloneqq \text{sample}(\mathcal{Q})$
    Initialise C as an empty list to store cardinality scores.
    for each $q \in \mathcal{Q}$ **do**
        $C.append(cs(q, \mathcal{R}))$
    end for
    $O.append([1 - \max(C)])$
end for

For our sample of $N \coloneqq 50$ excerpts from Haydn, Mozart, and Beethoven string quartets, the mean originality was $\text{mean}(OS) = 0.699$, with bootstrap 95%-confidence interval 0.672 and 0.725. We interpret this to mean that for the current corpus and sample (space of fast, first-movement Classical string quartets), composers wrote music that was 69.9% original, at least according to the note-counting music similarity measure employed here.

[5] See https://osf.io/96emr/ for the dataset, algorithms, and analyses.

4.2 Is This Algorithm's Output Sufficiently Original?

We can use the mean and confidence interval calculated above to help address the question of whether an algorithm's output is sufficiently original. Let us suppose we have a passage generated by an algorithm, and we traverse that output, collecting n-beat excerpts with 50% overlap, say, into a query set $\mathcal{Q}$. In this paper, we use $n \coloneqq 8, 16$ beats, which corresponds to 2- and 4-bar excerpts in 4-4 time, respectively. It is advisable to use at least two different window sizes, to probe the assumption that originality should increase with window size. In other words, different window sizes can be used to determine whether a worrisome-looking instance of low originality at the 2-bar level increases – and so becomes less worrisome – at a longer 4-bar window size.

We ran the MAIA Markov [6] and Music Transformer algorithms [16] to explore this question of sufficient originality, based on the training data of 64 string quartet movements described above. Markov model was built on the representation shown in Fig. 1(e). The Music Transformer model's training data was augmented by transposing the original pieces in the range $[-5, 6]$ MIDI notes, and then we sliced these into subsequences of fixed size 2,048 for batch training, giving a training set of 4,128 and a validation dataset of 564 subsequences. The model, with six layers, eight heads, and hidden size of 512, was trained with smoothed cross entropy loss [23] and the Adam optimiser [18] with custom learning rate schedule [2]. In keeping with the standard approach, the training process was stopped at epoch (checkpoint) 3, where the validation loss reached a minimum value of 1.183. Afterwards, 30 excerpts were generated by MAIA Markov and Music Transformer to form the query set $\mathcal{Q}$, based on which the mean value of originality scores were obtained by following the same method in Section 3.1, now for each time window as the excerpt is traversed.

For both algorithms, we see in Figs. 3(a) and (b) that the originality at the 2-bar level is low relative to the mean and 95%-confidence interval for the baseline, but this is to be expected because the baseline was calculated at the 4-bar level. What we expect to see is the solid line – indicating algorithm originality at the 4-bar level – lie entirely inside that confidence interval. This is the case for MAIA Markov [6], but Music Transformer's [16] mean originality level is entirely below this confidence interval, indicating it has issues with borrowing too heavily from the input on which it is trained.

Three typical worst case examples of copying are shown in Figs. 3(c), (d), and (e), with generated outputs on the left and original excerpts on the right. Fig. 3(c) shows one from MAIA Markov having 42.9% originality associated with Beethoven's String quartet no.6 in B-flat major, op.18, mvt.1, bars 61-64. And then, Figs. 3(d) shows one generated by the "best" checkpoint (checkpoint 3 with the minimum validation loss) of Music Transformer having 48.8% originality associated with Mozart's String quartet no.13 in D minor, K.173, mvt.1, bars 125-128. We found most of the generated outputs in this stage with less than 50% originality are due to repeating the same note, which is also frequently found in Classical string quartets, and the model tends to start by reproducing this simple kind of pattern. Finally, Figs. 3(e) shows one generated by checkpoint 15

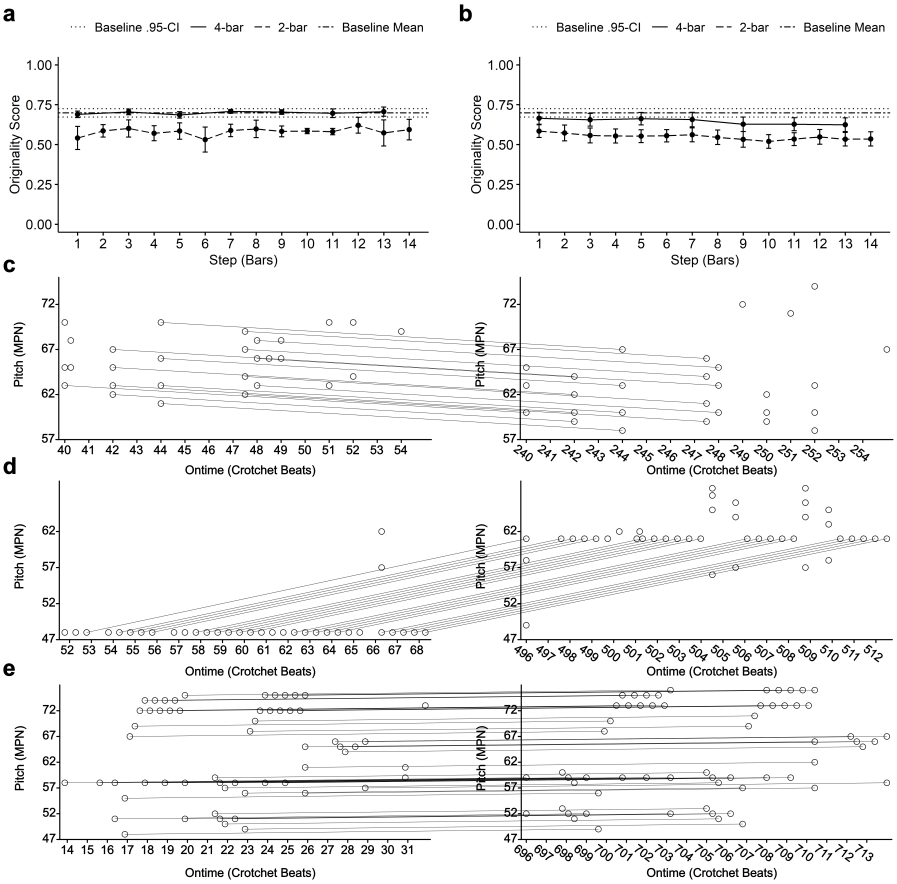

Fig. 3. Originality report for the MAIA Markov and Music Transformer algorithms. (a) and (b) show the change in originality scores over the course of the excerpts obtained for MAIA Markov and Music Transformer, respectively, at 2- and 4-bar levels compared to the baseline mean and 95%-confidence interval; (c), (d) and (e) show worst-case examples of copying by MAIA Markov and Music Transformer at checkpoints 3 and 15, respectively, where the generated outputs are on the left and the human-composed excerpts are on the right.

of Music Transformer having 9.9% originality associated with Beethoven's String quartet no. 1 in F major, op.18, mvt.1, bars 233-234. Generally, the model appears to be over-fit at this checkpoint. We infer from these originality reports and basic musicological interpretations that the results generated by Music Transformer gradually morph during training from reproduction of simple patterns (e.g., repeated notes) to verbatim use of more distinctive note sequences.

4.3 Incorporating Originality Reports into an Algorithmic Process: Originality Decreases as Epoch Increases

Here, we demonstrate the use of an originality report in the modelling process itself, as a means of analysing changes in originality as a function of model training or validation epoch. Music Transformer was used as an example of a deep learning model, with the train/validation split as in Section 4.1. To monitor the originality change along with the training process, 10 checkpoints including the initial point were saved. Again, we used each of them to generate 30 excerpts, to which the aforementioned originality report was applied. Afterwards, we calculated the mean value of those 30 originality scores for each checkpoint.

Fig. 4(a) and (b) show the change of loss and accuracy respectively over training. As mentioned previously, the standard training process would stop at epoch (checkpoint) 3, where the validation loss reaches a minimum, but we extended the training process further to more fully investigate the effect of training on originality. Fig. 4(c) and (d) contain a dashed line indicating the baseline originality level of 0.699 for the string quartet dataset. In Fig. 4(c), mean originality score decreases as a function of model training epoch, but remains largely in the 95%-confidence interval of the baseline originality level of the corpus. Fig. 4(d) is more concerning, indicating that minimum originality score decreases to well below the 95%-confidence interval of the baseline originality level of the corpus. Originality decreases until epoch 3, and then it stays relatively flat afterwards. However, as with the discussion of Figs. 3(c) and (d) in the previous subsection, we found that the model's borrowing still becomes more verbatim (or distinctive) after epoch 3, thus originality in a more general sense is still decreasing, a fact that is not immediately evident from Fig. 3 because the cardinality score does not consider distinctiveness, discussed further below.

5 Discussion

This paper puts forward the notion that AI for music generation should result in outputs that imitate instead of merely copying original pieces, and highlights that checks of whether this is the case – what we refer to as the originality report – are often omitted. We introduce the methodology of the originality report for baselining and evaluating the extent to which a generative model copies from training data. By substituting in different similarity metrics, it would be possible to adapt the methodology to have emphases on different musical dimensions, but here we take a relatively straightforward note-counting approach based on the

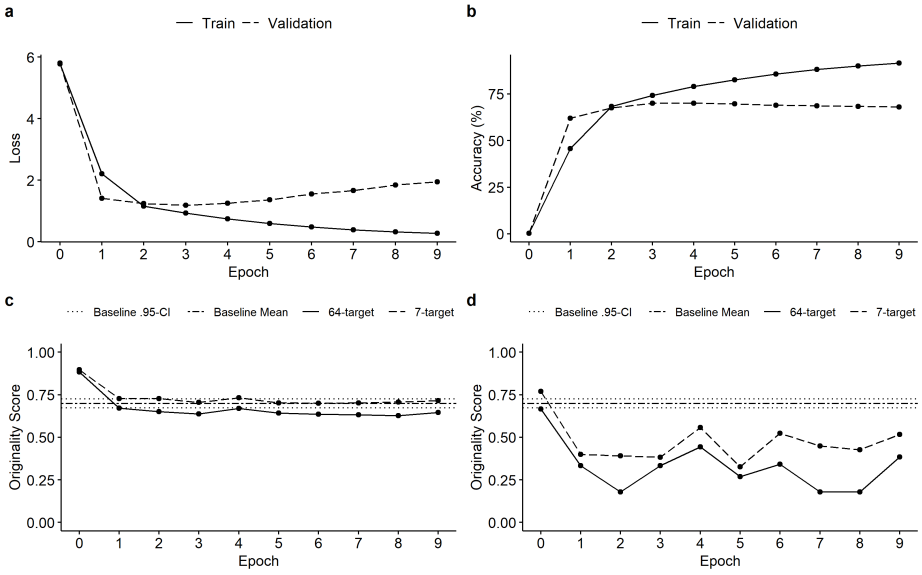

Fig. 4. (a) the loss curve of train and validation; (b) the accuracy curve of train and validation; (c) the mean originality curve for 64-target and 7-target sets; (d) the minimum originality score curve for 64-target and 7-target sets.

cardinality score [35,5]. We analyse outputs from two example models, one deep learning algorithm called Music Transformer [16] and one non-deep learning model called MAIA Markov [6], to illustrate the use of this methodology and the existence of music plagiarism in recent research.

We recognise Google Magenta for making their source code (e.g., for Music Transformer) publicly available, because it enables a level of scrutiny that has not always been possible for previous work in this field [27]. That said, the results indicate a phenomenon wherein this type of deep learning language model gradually copies increasingly distinctive chunks from pieces in the training set, calling into question whether it really *learns* to generate. More recent research found the information in training data can be retrieved from large language models, which highlights various issues of memorization [3]. Furthermore, using the conventional stopping criteria for the training process, the "best" model not only has a low level of originality, but also the quality of generated excerpts is low in the sense that the same note is repeated most of time (see Fig. 3(d)). Going forward, the field of deep learning needs to reconsider in what situations the conventional stopping criteria are appropriate: perhaps loss and accuracy should no longer be the only criteria when evaluating the model, because we need to prevent these models copying training data, especially when they are used increasingly in a "black-box" manner by practising musicians.

5.1 Limitations

The size of the dataset used above is smaller than that used for the original work on Music Transformer [16], and it is also quantised to a smaller set of time values. A potential solution is to pretrain the model with a large dataset to gain "general musical knowledge" and then finetune with a smaller style-specific dataset [11,9]. However, our dataset still represents a substantial amount of Classical music, and certainly enough to give a human music student an idea of the intended style, so if deep learning algorithms cannot operate on datasets of this size, then it should be considered a weakness of the deep learning approach rather than a limitation of our methodology.

The simplicity of the cardinality score has some appeal, but as noted in the review of existing work, it can mean that subtle variations along some musical dimension destroy any translational equivalence, giving a low cardinality score that is at odds with high perceived similarity. For instance, the expressive timing in the MAESTRO dataset [15] constitutes such subtle variations along the dimension of ontime, and make it ill-advised to use the cardinality score to assess the originality of an algorithm trained on these data. In addition, the cardinality score shares some general advantages of the geometric approach. But in its current use, it is also not able to take into account the distinctiveness of excerpts being compared [8,4]. For instance, Fig. 2 indicated an instance of similarity between Mozart and Haydn, but when we take into account how many Classical pieces end in this way, it is not a particularly distinctive or interesting example.

5.2 Future Work

We would like to see the originality report method that we have developed be embedded into the training processes of various music generation algorithms, to play a role as an advanced stopping criterion. Meanwhile, we will need to ensure that this criterion can still maintain the generalisability asserted by standard stopping criteria. Additionally, we will investigate the compatibility of the originality report method with model selection, which is often conducted as an outer loop of model training. Loss function engineering is a topic addressed in recent novel generating strategies (e.g., [13]), so it should be possible to merge high/low originality scores as rewards/penalties in training loss, to further investigate the problem that we have identified of language-based deep learning models appearing to be little more than powerful memorisers.

We will also explore the weighting of shift errors [5], a fingerprinting approach [1], and distinctiveness [8] to address the limitations mentioned above, arising from the simplicity of the cardinality score. This should mean originality reports can be generated for an algorithm trained on *any* music data, including those with expressive timing information, and taking into account distinctiveness with respect to an underlying corpus.

References

1. Arzt, A., Böck, S., Widmer, G.: Fast identification of piece and score position via symbolic fingerprinting. In: ISMIR. pp. 433–438 (2012)
2. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: Neural networks: Tricks of the trade, pp. 437–478. Springer (2012)
3. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. arXiv preprint arXiv:2012.07805 (2020)
4. Collins, T., Arzt, A., Frostel, H., Widmer, G.: Using geometric symbolic fingerprinting to discover distinctive patterns in polyphonic music corpora. In: Computational Music Analysis, pp. 445–474. Springer (2016)
5. Collins, T., Böck, S., Krebs, F., Widmer, G.: Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In: 53rd International Conference: Semantic Audio. Audio Engineering Society (2014)
6. Collins, T., Laney, R.: Computer-generated stylistic compositions with long-term repetitive and phrasal structure. Journal of Creative Music Systems **1**(2) (2017). https://doi.org/https://doi.org/10.5920/JCMS.2017.02
7. Collins, T., Laney, R., Willis, A., Garthwaite, P.H.: Developing and evaluating computational models of musical style. AI EDAM **30**(1), 16–43 (2016)
8. Conklin, D.: Discovery of distinctive patterns in music. Intelligent Data Analysis **14**(5), 547–554 (2010)
9. Conklin, D., Witten, I.H.: Multiple viewpoint systems for music prediction. Journal of New Music Research **24**(1), 51–73 (1995)
10. Cope, D.: Computer models of musical creativity. MIT Press Cambridge (2005)
11. Donahue, C., Mao, H.H., Li, Y.E., Cottrell, G.W., McAuley, J.: LakhNES: Improving multi-instrumental music generation with cross-domain pre-training. In: ISMIR. pp. 685–692 (2019)
12. Dong, H.W., Hsiao, W.Y., Yang, L.C., Yang, Y.H.: Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
13. Elgammal, A., Liu, B., Elhoseiny, M., Mazzone, M.: Can: Creative adversarial networks generating "art" by learning about styles and deviating from style norms. In: 8th International Conference on Computational Creativity, ICCC 2017. Georgia Institute of Technology (2017)
14. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. pp. 6645–6649. IEEE (2013)
15. Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.Z.A., Dieleman, S., Elsen, E., Engel, J., Eck, D.: Enabling factorized piano music modeling and generation with the MAESTRO dataset. In: International Conference on Learning Representations (2019)
16. Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D.: Music transformer: Generating music with long-term structure. In: ICLR (2018)
17. Janssen, B., Collins, T., Ren, I.Y.: Algorithmic ability to predict the musical future: Datasets and evaluation. In: ISMIR. pp. 208–215 (2019)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

19. Lewin, D.: Generalized musical intervals and transformations. Oxford University Press, USA (2007), originally published by Yale University Press, New Haven, 1987
20. Meredith, D.: The ps13 pitch spelling algorithm. Journal of New Music Research **35**(2), 121–159 (2006)
21. Meredith, D., Lemström, K., Wiggins, G.A.: Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. Journal of New Music Research **31**(4), 321–345 (2002)
22. Müllensiefen, D., Pendzich, M.: Court decisions on music plagiarism and the predictive value of similarity algorithms. Musicae Scientiae **13**(1_suppl), 257–295 (2009)
23. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: Advances in Neural Information Processing Systems. pp. 4694–4703 (2019)
24. Neely, A.: Why the Katy Perry/Flame lawsuit makes no sense. https://www.youtube.com/watch?v=0ytoUuO-qvg, accessed: 2020-10-30
25. Oore, S., Simon, I., Dieleman, S., Eck, D., Simonyan, K.: This time with feeling: Learning expressive musical performance. Neural Computing and Applications pp. 1–13 (2018)
26. Otzen, E.: Six seconds that shaped 1,500 songs. https://www.bbc.co.uk/news/magazine-32087287, accessed: 2020-10-30
27. Pachet, F., Papadopoulos, A., Roy, P.: Sampling variations of sequences for structured music generation. In: ISMIR. pp. 167–173 (2017)
28. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
29. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D.: A hierarchical latent vector model for learning long-term structure in music. In: International Conference on Machine Learning. pp. 4364–4373. PMLR (2018)
30. Ruiz, A., Simon, I.: My only problem with Magenta's Transformer. Magenta Discuss Google Group, https://groups.google.com/a/tensorflow.org/g/magenta-discuss/c/Oxiq-Gdaavk/m/uHIsQZKtBwAJ, accessed: 2020-10-30
31. Selfridge-Field, E.: Conceptual and representational issues in melodic comparison. Computing in Musicology (1999)
32. Stav, I.: Musical plagiarism: a true challenge for the copyright law. DePaul J. Art Tech. & Intell. Prop. L **25**, 1 (2014)
33. Sturm, B.L., Iglesias, M., Ben-Tal, O., Miron, M., Gómez, E.: Artificial intelligence and music: open questions of copyright law and engineering praxis. In: Arts. vol. 8, p. 115. Multidisciplinary Digital Publishing Institute (2019)
34. Todd, P.M.: A connectionist approach to algorithmic composition. Computer Music Journal **13**(4), 27–43 (1989)
35. Ukkonen, E., Lemström, K., Mäkinen, V.: Geometric algorithms for transposition invariant content-based music retrieval. In: Proceedings of the International Symposium on Music Information Retrieval. pp. 193–199 (2003)
36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
37. Volk, A., Chew, E., Margulis, E.H., Anagnostopoulou, C.: Music similarity: Concepts, cognition and computation. J. New Music Research **45**(3), 207–209 (2016)
38. Wang, A.L.C., Smith III, J.O.: System and methods for recognizing sound and music signals in high noise and distortion (2012), patent US 8,190,435 B2. Continuation of provisional application from 2000.
39. Yates, P.: Twentieth century music: Its evolution from the end of the harmonic era into the present era of sound. Allen & Unwin (1968)